Calculation Time for Svx Reconstruction

Svx Meeting
July 4, 2008

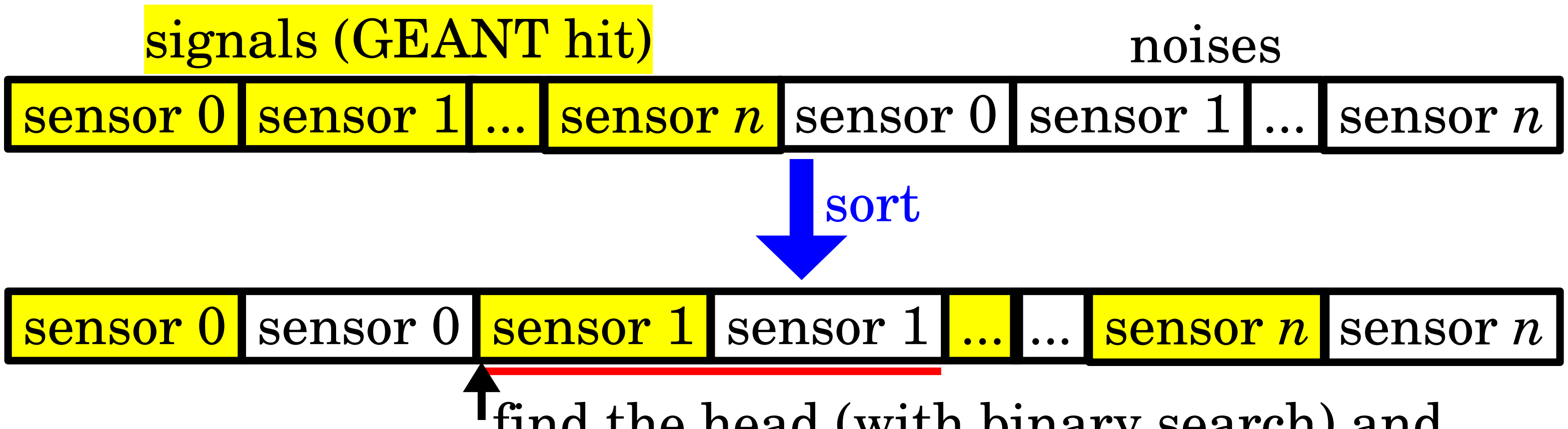
Kenichi Nakano
RIKEN

D. Winter, Manabu & My Observations

- D. Winter found in his single-particle simulation that
  - it got stuck (it was because the geometry in database was old)
  - it was very slow
- We checked calculation times that each section of svx reconstruction uses
  - getrusage (resource usage) is used (SvxCommon::AddRUsage() etc. in CVS)
  - in single-particle events
    - sorting rawhit list takes 60 sec/evt with 55k (>+1σ) noise hits
    - 1.3 sec/evt 8.6k (>+2σ) noise hits
    - >95% of total time per event in case of 55k noise hits
    - time for sort $T \sim O(n^2)$ ... worst case of qsort
  - in central Au+Au events
    - sorting rawhit list takes 1.5~2.0 sec/evt with 55k (>+1σ) noise hits
    - 0.5~1.0 sec/evt 8.6k (>+2σ) noise hits
    - <30% of total time per event
  - rawhit-ADC threshold can be 21 (+2σ of noise dist.), but still too slow

Why & How SvxRawhitList is Sorted

- Why it has to be sorted



signals (GEANT hit)

noises

sensor 0 | sensor 1 | ... | sensor *n* | sensor 0 | sensor 1 | ... | sensor *n*

sort

sensor 0 | sensor 0 | sensor 1 | sensor 1 | ... | ... | sensor *n* | sensor *n*

find the head (with binary search) and
use continuous hits in clustering on sensor 1

- How it was sorted
  - comparison with elements: (layer, ladder, sensor, section, readout, chan.)
  - TClonesArray::Sort() –> TSeqCollection::QSort()
    - http://root.cern.ch/root/html516/src/TSeqCollection.cxx.html#HZ.gUE
    - it is really bad. it stirs a sorted list like { 1, 1, 1, 2, 3, 4 }!?
- How it is sorted now
  - sorting with (section, readout, chan.) is not needed. now they are ignored
  - make signal+noise hits at once for each sensor –> no need to sort
  - sorting a sorted list mustn't spend time, but ROOT QSort does. I commented out the lines that call SvxRawhitList::Sort()

Calculation Time

- With updated codes

	Au+Au		single particle	
	8.6k noise	no noise	8.6k noise	no noise
get & fill GEANT hits	0.16 sec	0.14	-	-
makeRawhits() + AddNoise()	0.13	0.08	0.06	-
pixel clustering	0.10	0.09	0.02	-
stripixel clustering	0.22	0.20	0.02	-
total in process_event()	0.90	0.80	0.12	-
others / event	2~4	2~4	0.10	0.02

  - have been optimized. hard to make it more faster
  - committed to CVS on July 3

- When SvxRawhitList::Sort() turns on with 8.6k noise
  - in Au+Au, sorting takes ~2 sec and process_event() takes 2~3 sec
  - in single particle, sorting takes 0.22 sec and process_event() takes 0.22 sec

Remarks

- The proper order of rawhits in SvxRawhitList is now a responsibility to coding algorithm. Beware when modifying.
  - don't use SvxRawhitList::sort_sensorID()
  - the code also assumes, from the 1st revision, that the rawhit list is in hit-ID order without sort when finding an unused hit-ID

- The replacement of TClonesArray in SvxHitList with STL vector (or deque) should recover a reasonable sortability. But it is a huge modification.

- During this study, I found a minor bug in SvxStripixel::makeRaw(). It sometimes failed in merging two GEANT hits into one rawhit.